

```

report   zvik_oo_alv_all.
data : o_container type ref to cl_gui_custom_container,
      o_grid       type ref to cl_gui_alv_grid,
      it_ekko       type table of ekko.

class lcl1_event_receiver definition.
  public section.
    methods : handle_double_click1 for event double_click
              of cl_gui_alv_grid importing e_row e_column,

              handle_toolbar for event toolbar
              of cl_gui_alv_grid importing e_object,

              handle_user_command for event user_command
              of cl_gui_alv_grid importing e_ucomm.

endclass.

*Implement Event Handler Class
class lcl1_event_receiver implementation.

method handle_user_command.
  data it_ekpo type table of   ekpo.
  case e_ucomm.
    when 'GETAI'.
      if it_ekko is not initial .
        select * into table it_ekpo from ekpo for all entries i
n it_ekko
                                                    where ebeln = it_ekko-
ebeln.
        call function 'REUSE_ALV_GRID_DISPLAY'
          exporting
            i_structure_name          = 'EKPO'
            tables
              t_outtab                = it_ekpo .
          endif.
        endcase.

```

```
endmethod.
```

```
method handle_toolbar.
```

```
data wa_button type line of ttb_button.
```

```
wa_button-function = 'GETAI'.
```

```
wa_button-text = 'GetAllItems'.
```

```
append wa_button to e_object->mt_toolbar.
```

```
endmethod.
```

```
method handle_double_click1.
```

```
data : wa_ekko type ekko,
```

```
it_ekpo type table of ekpo.
```

```
read table it_ekko into wa_ekko index e_row-index.
```

```
if e_column = 'EBELN'.
```

```
select * into table it_ekpo from ekpo
```

```
where ebeln = wa_ekko-ebeln.
```

```
call function 'REUSE_ALV_GRID_DISPLAY'
```

```
exporting
```

```
i_structure_name = 'EKPO'
```

```
tables
```

```
t_outtab = it_ekpo
```

```
.
```

```
endif.
```

```
endmethod.
```

```
endclass.
```

```
data o_event_rcvr type ref to lcl1_event_receiver.
```

```
start-of-selection.
```

```
call screen 9000.
```

```
module display_alv_9000 output.
```

```
create object o_container
```

```
exporting
```

```

        container_name                = 'C1' .

create object o_grid
exporting

        i_parent                      = o_container .

create object o_event_rcvr.

*For Each Event
set handler : o_event_rcvr->handle_double_click1      for o_gri
d,
                o_event_rcvr->handle_toolbar          for o_grid,
                o_event_rcvr->handle_user_command      for o_grid.

call method o_grid->set_table_for_first_display
exporting
        i_structure_name              = 'EKK0'
changing
        it_outtab                     = it_ekko .

endmodule.

module read_data_9000 output.
        select * into table it_ekko from ekko up to 20 rows
                where bstyp = 'F'.
endmodule.

module status_9000 output.
        set pf-status 'ZSTATUS'.
endmodule.

module user_command_9000 input.
        case sy-ucomm.
                when 'BK'.
                        leave to screen 0.

```



```

        handle_top_of_page for event top_of_page
                                of cl_gui_alv_grid,
        handle_before_user_command for event
        before_user_command of cl_gui_alv_grid
        importing e_ucomm .

```

```

endclass.

```

**Implement Event Handler Class*

```

class lcl_event_receiver implementation.

```

```

method handle_before_user_command.

```

```

data :   it_index_rows type lvc_t_row,
        wa              type line of lvc_t_row,
        it_selected_ekko type table of ekko,
        wa_ekko         type ekko.

```

```

case e_ucomm.

```

```

    when '&PC'.

```

```

        message 'I am Executed before U' type 'I'.

```

**Returns Only the Indexs of the Selected Rows*

```

call method o_grid->get_selected_rows

```

```

    importing

```

```

        et_index_rows = it_index_rows

```

```

*   et_row_no      =

```

```

    .

```

**Get the Content by reading for Each Index In it_index_rows*

```

loop at it_index_rows into wa.

```

```

    read table it_ekko into

```

```

    wa_ekko index wa-index.

```

```

    append wa_ekko to it_selected_ekko.

```

```

endloop.

```

```

call method cl_gui_frontend_services=>gui_download

```

```

    exporting

```

```

*   bin_filesize      =

```

```

    filename           = 'F:\7AM\ALV\EKKO_SELECTED.TXT'

```

```

*   filetype          = 'ASC'

```

```

*      append                        = SPACE
*      write_field_separator        = SPACE
*      header                      = '00'
*      trunc_trailing_blanks       = SPACE
*      write_lf                    = 'X'
*      col_select                  = SPACE
*      col_select_mask             = SPACE
*      dat_mode                    = SPACE
*      confirm_overwrite           = SPACE
*      no_auth_check               = SPACE
*      codepage                    = SPACE
*      ignore_cerr                 = ABAP_TRUE
*      replacement                 = '#'
*      write_bom                   = SPACE
*      trunc_trailing_blanks_eol   = 'X'
*      wkl_n_format                = SPACE
*      wkl_n_size                  = SPACE
*      wkl_t_format                = SPACE
*      wkl_t_size                  = SPACE
*      show_transfer_status        = 'X'
*      fieldnames                  =
*      write_lf_after_last_line    = 'X'
*      IMPORTING
*      filelength                  =
changing
      data_tab                    = it_selected_ekko
*      EXCEPTIONS
*      file_write_error            = 1
*      no_batch                    = 2
*      gui_refuse_filetransfer     = 3
*      invalid_type                = 4
*      no_authority                = 5
*      unknown_error               = 6
*      header_not_allowed          = 7
*      separator_not_allowed       = 8
*      filesize_not_allowed        = 9
*      header_too_long             = 10
*      dp_error_create             = 11
*      dp_error_send               = 12
*      dp_error_write              = 13

```

```

*      unknown_dp_error          = 14
*      access_denied             = 15
*      dp_out_of_memory          = 16
*      disk_full                 = 17
*      dp_timeout                = 18
*      file_not_found            = 19
*      dataprovider_exception    = 20
*      control_flush_error       = 21
*      not_supported_by_gui      = 22
*      error_no_gui              = 23
*      others                    = 24

```

```

if sy-subrc = 0.
    message 'selected Lines are Successfully Downloaded' type 'S'
.
endif.

uline.

```

```

    call method o_grid->set_user_command
    exporting
        i_ucomm = space "Std.Function Deactivated
.

```

```

endcase.
endmethod .

```

```

method handle_top_of_page.
call method o_document->add_picture
exporting
    picture_id          = 'ENJOYSAP_LOGO'
*      width            =
*      alternative_text =
*      tabindex         =
.

```

```

call method o_document->add_text
exporting
    text          = 'List Of Purchasing Documents'
*      text_table =

```

```

*   fix_lines      =
*   sap_style      =
*   sap_color      =
*   sap_fontsize   =
*   sap_fontstyle  =
*   sap_emphasis   =
*   style_class    =
*   CHANGING
*   document       =

```

```

call method o_document->add_picture
exporting

```

```

    picture_id      = 'ENJOYSAP_LOGO'
*   width           =
*   alternative_text =
*   tabindex        =

```

```

call method o_document->display_document
exporting

```

```

*   reuse_control   =
*   reuse_registration =
*   container       =
    parent          = o_header
*   EXCEPTIONS
*   html_display_error = 1
*   others             = 2

```

```

if sy-subrc <> 0.

```

```

* MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
*           WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.

```

```

endif.

```

```

endmethod.

```

```

method handle_user_command.

```

```

    data it_ekpo type table of ekpo.

```

```

    case e_ucomm.

```

```

        when 'GETAI'.

```

```

        if it_ekko is not initial .

```



```

select * into table it_ekpo from ekpo for all entries i
n it_ekko
where ebeln = it_ekko-
ebeln.

```

```

call function 'REUSE_ALV_GRID_DISPLAY'
exporting

```

```

*          I_INTERFACE_CHECK                = ' '
*          I_BYPASSING_BUFFER                = ' '
*          I_BUFFER_ACTIVE                  = ' '
*          I_CALLBACK_PROGRAM                = ' '
*          I_CALLBACK_PF_STATUS_SET          = ' '
*          I_CALLBACK_USER_COMMAND           = ' '
*          I_CALLBACK_TOP_OF_PAGE            = ' '
*          I_CALLBACK_HTML_TOP_OF_PAGE       = ' '
*          I_CALLBACK_HTML_END_OF_LIST       = ' '
*
*          i_structure_name                  = 'EKPO'
*          I_BACKGROUND_ID                  = ' '
*          I_GRID_TITLE                      =
*          I_GRID_SETTINGS                   =
*          IS_LAYOUT                         = IS_LAYOUT
*          IT_FIELDCAT                       =
*          IT_EXCLUDING                      =
*          IT_SPECIAL_GROUPS                 =
*          IT_SORT                           =
*          IT_FILTER                         =
*          IS_SEL_HIDE                       =
*          I_DEFAULT                         = 'X'
*          I_SAVE                            = ' '
*          IS_VARIANT                        =
*          IT_EVENTS                         = IT_EVENTS
*          IT_EVENT_EXIT                     =
*          IS_PRINT                          =
*          IS_REPREP_ID                      =
*          I_SCREEN_START_COLUMN             = 0
*          I_SCREEN_START_LINE               = 0
*          I_SCREEN_END_COLUMN               = 0
*          I_SCREEN_END_LINE                 = 0
*          I_HTML_HEIGHT_TOP                 = 0
*          I_HTML_HEIGHT_END                 = 0
*          IT_ALV_GRAPHICS                   =

```

```

*          IT_HYPERLINK                      =
*          IT_ADD_FIELDCA1                   =
*          IT_EXCEPT_QINFO                 =
*          IR_SALV_FULLSCREEN_ADAPTER        =
*      IMPORTING
*          E_EXIT_CAUSED_BY_CALLER           =
*          ES_EXIT_CAUSED_BY_USER           =

      tables
          t_outtab                          = it_ekpo
*          EXCEPTIONS
*          PROGRAM_ERROR                    = 1
*          OTHERS                          = 2
      .
      if sy-subrc <> 0.
*  MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
*          WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
      endif.
    endif.
  endcase.
endmethod.

method handle_toolbar.
  data wa_button type line of ttb_button.
    wa_button-function = 'GETAI'.
    wa_button-text = 'GetAllItems'.
    append wa_button to e_object->mt_toolbar.

endmethod.

method handle_double_click.
data : wa_ekko type ekko,
      it_ekpo type table of ekpo.
read table it_ekko into wa_ekko index e_row-index.

if e_column = 'EBELN'.
  select * into table it_ekpo from ekpo
        where ebeln = wa_ekko-ebeln.

    call function 'REUSE_ALV_GRID_DISPLAY'
      exporting

```

```

*      I_INTERFACE_CHECK                      = ' '
*      I_BYPASSING_BUFFER                    = ' '
*      I_BUFFER_ACTIVE                      = ' '
*      I_CALLBACK_PROGRAM                   = ' '
*      I_CALLBACK_PF_STATUS_SET             = ' '
*      I_CALLBACK_USER_COMMAND              = ' '
*      I_CALLBACK_TOP_OF_PAGE               = ' '
*      I_CALLBACK_HTML_TOP_OF_PAGE          = ' '
*      I_CALLBACK_HTML_END_OF_LIST          = ' '
i_structure_name                           = 'EKPO'
*      I_BACKGROUND_ID                     = ' '
*      I_GRID_TITLE                        =
*      I_GRID_SETTINGS                     =
*      IS_LAYOUT                           =
*      IT_FIELDCAT                         =
*      IT_EXCLUDING                        =
*      IT_SPECIAL_GROUPS                   =
*      IT_SORT                             =
*      IT_FILTER                           =
*      IS_SEL_HIDE                         =
*      I_DEFAULT                           = 'X'
*      I_SAVE                              = ' '
*      IS_VARIANT                           =
*      IT_EVENTS                           = IT_EVENTS
*      IT_EVENT_EXIT                       =
*      IS_PRINT                             =
*      IS_REPREP_ID                        =
*      I_SCREEN_START_COLUMN                = 0
*      I_SCREEN_START_LINE                  = 0
*      I_SCREEN_END_COLUMN                  = 0
*      I_SCREEN_END_LINE                    = 0
*      I_HTML_HEIGHT_TOP                    = 0
*      I_HTML_HEIGHT_END                    = 0
*      IT_ALV_GRAPHICS                      =
*      IT_HYPERLINK                        =
*      IT_ADD_FIELDCAT                      =
*      IT_EXCEPT_QINFO                     =
*      IR_SALV_FULLSCREEN_ADAPTER            =
IMPORTING
*      E_EXIT_CAUSED_BY_CALLER              =

```

```

*          ES_EXIT_CAUSED_BY_USER          =
      tables
      t_outtab          = it_ekpo
*          EXCEPTIONS
*          PROGRAM_ERROR          = 1
*          OTHERS          = 2

```

```

      .
      if sy-subrc <> 0.
* MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
* WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
      endif.
      endif.

```

```

endmethod.
endclass.

```

```

data o_event_rcvr type ref to lcl_event_receiver.
start-of-selection.
  call screen 9000.

```

```

*&-----
-----*
*&      Module  DISPLAY_ALV_9000  OUTPUT
*&-----
-----*

```

```

module display_alv_9000 output.

```

```

data is_layout type lvc_s_layo.

```

```

is_layout-sel_mode = 'D'. "Allows Multi Line Selection

```

```

create object o_grid
exporting

```

```

*   i_shellstyle          = 0
*   i_lifetime           =
*   i_parent              = o_main
*   i_appl_events         = space
*   i_parentdbg           =
*   i_applogparent        =
*   i_graphicsparent      =

```

```

*      i_name          =
*      i_fcat_complete = space
*      EXCEPTIONS
*      error_cntl_create = 1
*      error_cntl_init   = 2
*      error_cntl_link   = 3
*      error_dp_create   = 4
*      others            = 5

```

```

if sy-subrc <> 0.

```

```

* MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
*           WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.

```

```

endif.

```

```

*Register the Event Handler Methods

```

```

create object o_event_rcvr.

```

```

*For Each Event

```

```

set handler : o_event_rcvr->handle_double_click      for o_grid,
              o_event_rcvr->handle_toolbar           for o_grid,
              o_event_rcvr->handle_user_command       for o_grid,
              o_event_rcvr->handle_top_of_page        for o_grid,
              o_event_rcvr->handle_before_user_command for o_gri
d.

```

```

create object o_document

```

```

.

```

```

*To Trigger the List Processing Event -> 'TOP_OF_PAGE'

```

```

call method o_grid->list_processing_events
exporting

```

```

    i_event_name      = 'TOP_OF_PAGE'
    i_dyndoc_id       = o_document
*      is_subtottxt_info =
*      ip_subtot_line   =
*      i_table_index   =

```

```

*   CHANGING
*   c_subtottxt      =
*   .

data : it_fieldcat type lvc_t_fcat,
      wa_fieldcat type line of lvc_t_fcat.

call function 'LVC_FIELDCATALOG_MERGE'
exporting
*   I_BUFFER_ACTIVE          =
*   i_structure_name         = 'EKKO'
*   I_CLIENT_NEVER_DISPLAY   = 'X'
*   I_BYPASSING_BUFFER       =
*   I_INTERNAL_TABNAME       =
changing
*   ct_fieldcat              = it_fieldcat
* EXCEPTIONS
*   INCONSISTENT_INTERFACE   = 1
*   PROGRAM_ERROR            = 2
*   OTHERS                   = 3
*   .

if sy-subrc <> 0.
*   Implement suitable error handling here
endif.

clear wa_fieldcat.
      wa_fieldcat-emphasize = 'C71'.
modify it_fieldcat from wa_fieldcat transporting emphasize
      where fieldname = 'LIFNR'.
call method o_grid->set_table_for_first_display
exporting
*   i_buffer_active          =
*   i_bypassing_buffer       =
*   i_consistency_check      =
*   i_structure_name         = 'EKKO'
*   is_variant               =
*   i_save                   =
*   i_default                 = 'X'
*   is_layout                 = is_layout
*   is_print                  =

```

```

*      it_special_groups                =
*      it_toolbar_excluding             =
*      it_hyperlink                     =
*      it_alv_graphics                  =
*      it_except_qinfo                  =
*      ir_salv_adapter                   =
changing
      it_outtab                         = it_ekko
      it_fieldcatalog                   = it_fieldcat
*      it_sort                          =
*      it_filter                        =
*      EXCEPTIONS
*      invalid_parameter_combination = 1
*      program_error                   = 2
*      too_many_lines                   = 3
*      others                           = 4
.
if sy-subrc <> 0.
* MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
*          WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
endif.

endmodule.                                " DISPLAY_ALV_9000  OUTPUT

*&-----
*-----*
*&      Module  READ_DATA_9000  OUTPUT
*&-----
*-----*
module read_data_9000 output.
  select * into table  it_ekko from ekko up to 20 rows
        where bstyp = 'F'.
endmodule.                                " READ_DATA_9000  OUTPUT

*&-----
*-----*
*&      Module  STATUS_9000  OUTPUT
*&-----
*-----*

```

```

module status_9000 output.
  set pf-status 'ZSTATUS'.
  * SET TITLEBAR 'xxx'.

endmodule.                                " STATUS_9000 OUTPUT

*&-----
-----*
*&      Module  USER_COMMAND_9000  INPUT
*&-----
-----*
module user_command_9000 input.
  case sy-ucomm.
    when 'BK'.
      leave to screen 0.
  endcase.
endmodule.                                " USER_COMMAND_9000 INPUT

*&-----
-----*
*&      Module  SPLIT_CONTAINER_9000  OUTPUT
*&-----
-----*
module split_container_9000 output.
create object o_container
exporting
*      parent                =
  container_name              = 'C1'
*      style                  =
*      lifetime               = lifetime_default
*      repid                  =
*      dynnr                  =
*      no_autodef_progid_dynnr =
*      EXCEPTIONS
*      cntl_error              = 1
*      cntl_system_error      = 2
*      create_error            = 3
*      lifetime_error          = 4
*      lifetime_dynpro_dynpro_link = 5
*      others                  = 6

```



```

.  

if sy-subrc <> 0.  

* MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO  

*           WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.  

endif.

```

```

create object o_splitter  

exporting  

*   link_dynnr      =  

*   link_repia      =  

*   shellstyle      =  

*   left            =  

*   top             =  

*   width           =  

*   height          =  

*   metric          = cntl_metric_dynpro  

*   align           = 15  

parent             = o_container  

rows               = 2  

columns           = 1  

*   no_autodef_progid_dynnr =  

*   name            =  

* EXCEPTIONS  

*   cntl_error      = 1  

*   cntl_system_error = 2  

*   others          = 3

```

```

.  

if sy-subrc <> 0.  

* MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO  

*           WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.  

endif.

```

```

call method o_splitter->get_container  

exporting  

    row      = 1  

    column   = 1  

receiving  

    container = o_header  

.

```

```
call method o_splitter->set_row_height
exporting
    id = 1 "Row1"
    height = 25
* IMPORTING
* result =
* EXCEPTIONS
* cntl_error = 1
* cntl_system_error = 2
* others = 3
```

```
* MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
*           WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
endif.
```

```
endmodule.           " SPLIT_CONTAINER_9000  OUTPUT
```